Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method of managing memory in a multi-threaded processing environment including respective local thread stacks and heaps and a global heap, said method comprising:

creating an object in a thread heap;

monitoring the object to determine whether the object is referenced only from a given thread stack;

~~assigning a status to the object; and~~

~~changing the status of the object under certain conditions~~

associating a local status with the object when the object is referenced only from a given thread stack;

changing the status of the object to a global status when the object is not referenced from only the given thread stack; and

deleting from a given thread heap one or more local objects when the object is associated with a local status.

2. (cancelled)

3. (previously presented) The method as claimed in claim 1 further comprising deleting from a given thread heap one or more local objects when they are not accessible from a local root.

4. (previously presented) The method as claimed in claim 3 where accessibility is determined by tracing from the local root.

5. (previously presented) The method as claimed in claim 4 wherein the status of an object in the given thread heap is changed to global if the object is assigned to a static variable or if the object is assigned to a field in any other object.

6. (previously presented) The method as claimed in claim 3 further comprising intercepting assignment operations to an object in the thread heap to assess whether the status should be changed.

7. (cancelled)

8. (cancelled)

9. (cancelled)

10. (cancelled)

11. (currently amended) The method as claimed in claim ~~10~~ 1 further comprising using multiples of two or more bytes in a thread heap to store the objects whereby there is at least one spare bit in the object length variable and using the at least one spare bit as the flag.

12. (previously presented) The method as claimed in claim 11 further comprising moving objects whose status is global from the thread heap to a global heap.

13. (previously presented) The method as claimed in claim 12 further comprising compacting the accessible local objects in a thread heap.

14. (previously presented) The method as claimed in claim 1 wherein certain objects are associated with a global status on creation.

15. (previously presented) The method as claimed in claim 14 where said certain objects include Class objects, Thread objects and Runnable objects.

16. (previously presented) The method as claimed in claim 14 further comprising a step of analysing whether an object is likely to be made global and associating such an object with a global status on creation.

17. (previously presented) The method as claimed in claim 16 further comprising allocating objects assigned as global on creation to the global heap.

18. (currently amended) A system for managing memory in a multi-threaded processing environment comprising:

respective local thread stacks and heaps;

a global heap;

means for creating an object in a thread heap;

means for monitoring the object to determine whether the object is referenced only from a given thread stack;

means for associating a local status with the object when the object is referenced only from a given thread stack; ~~and~~

means for changing the status of the object ~~under certain conditions~~ when the object is not referenced only from a given thread stack; and

means for deleting from the thread heap one or more local objects when the one or more objects are associated with a local status.

19. (cancelled)

20. (previously presented) The system as claimed in claim 18 further comprising means for deleting from the thread heap one or more local objects when they are not reachable from a local root.

21. (previously presented) The system as claimed in claim 20 further comprising:

means for changing the status of an object in the thread heap to global if the object is assigned to a static variable or if the object is assigned to a field in any other object.

22. (currently amended) A computer program product stored on a computer readable storage medium for managing memory in a multi-threaded processing environment including respective local thread stacks and heaps and a global heap, when executed on a computer, said product comprising:

instructions for creating an object in a thread heap;

instructions for monitoring whether the object is referenced only from a given thread stack; and

means for associating a status with the object, wherein the status is one of a local status or a global status;

means for changing the status of the object ~~under certain conditions~~ when the status is a local status.

23. (currently amended) ~~A~~ The product as claimed in claim 22 further comprising:

means for initially associating a local status with the object;

means for changing the status of the object to global under certain conditions.

24. (previously presented) The product as claimed in claim 22 further comprising means for deleting from the thread heap one or more local objects when they are not a local root.

25. (previously presented) The product as claimed in claim 24 where accessibility is determined by tracing from the local root.

26. (currently amended) The product as claimed in claim 25 wherein the local status of an object in the thread heap is changed to a global status if the object is assigned to a static variable or if the object is assigned to a field in any other object.

27. (Cancelled)